EXHIBIT G

Exhibit G

Summary of Opinions of Professor John Mitchell
In Support of Microsoft's Proposed Claim Constructions

1. In the field of computer security, terms such as "secure," "protect," and "tamper resistance" are understood differently depending on the particular context in which they are used. They have such a range of possible meanings that context is essential to understanding what these terms mean in a given instance. The same is true for terms like "govern" and "control" when they are used to describe computer systems or access to information.

A person skilled in the computer security field would not expect to use a dictionary to understand what these terms mean in a given context; rather, he or she would expect to review the particular reference or system in question to see what adversarial events or attacks are being defended against. Generally speaking, dictionary "definitions" are not sufficient for understanding how these terms are meant in a particular case. A number of terms and phrases used in the February 1995 application (such as "VDE," "PPE," and "secure container") are also not likely to be found in dictionaries.

2. The February 1995 application (which is sometimes referred to as the "Big Book") never clearly explains what it means by "security." It would not be clear to someone of average skill in the field what "secure" means in that application -- for example, with regard to systems, system components, information, or processes. The same is true for such terms as "protected" and "tamper resistant."

3. If a reasonably skillful computer security professional were to presume that "secure" has all of the attributes that are promised in the February 1995 application, then "secure" requires a guarantee of secrecy, authenticity, integrity, nonrepudiation, and availability, against all security threats identified in that application other than excessively costly brute

force attacks. (What constitutes excessive cost in this context is not clearly explained). Again taking the February 1995 application's promises for context, "tamper resistance" requires that some barrier is in place which prevents access to or alteration of information in an unauthorized manner. The terms "secure" and "security", and additional terms such as "secure container," "control," "govern," "protect," "protected processing environment," "host processing environment" and "virtual distribution environment," would be understood, to the extent possible, as set forth in Microsoft's PLR 4-2 Statement, as opposed to the definitions listed in InterTrust's PLR 4-2 Statement.

4. Professor Mitchell will explain the qualifications of a person of reasonable skill in the computer security field, including as of February 13, 1995, and explain how cited references (such as U.S. Patent 5,634,012 to Stefik et al., U.S. Patents 4,868,877 and 5,337,360 to Fischer, Choudhury et al.'s "Copyright Protection for Electronic Publishing over Computer Networks," U.S. Patent 4,658,093 to Hellman, and Mori et al.'s "Superdistribution: The Concept and Architecture" (Transactions of the IECE 1990)) would influence such a person's understanding of the InterTrust disclosure. He may also address the substance of additional references published or created before February 13, 1995, not cited in the InterTrust patents.

5. The specifications of the '721, '900, and '861 patents do not resolve any of these problems with the Big Book application.

Summary of Opinions of Professor David Maier
in Support of Microsoft's Proposed Claim Constructions

1. The specification of U.S. Patent No. 6,253,193 ("the '193 patent") describes several mandatory features of the Virtual Distribution Environment ("VDE") architecture, including:

- the creation of a comprehensive data security and commerce world;
- the ability to handle all types of digital works independent of computing platform, making it a single, general purpose solution in contrast to multiple, limited purpose solutions;
- flexible control mechanisms that can be applied to any granularity of content;
- control mechanisms that are configurable by any user, not just the system designers or content providers; and
- isolation of the system programs and protected works from the non-VDE world, preventing observation, alteration, interference, or removal from the VDE, except as permitted by the VDE control mechanisms.

This does not mean that the capabilities of the Virtual Distribution Environment can be achieved, only that these are features that the '193 patent makes clear a VDE must have.

2. The specification of the '193 patent describes a system that requires several architectural elements including at least the following:

- VDE Foundation Hardware and Software – installed throughout an infrastructure of interlinked computing devices;
- The VDE "Secure Container" – a mechanism for packaging protected works, control information, and administrative information; and

- The VDE "Control" – a mechanism for defining the regimen for using protected information that is inside a secure container.

3. Professor Maier will describe the background of a person of ordinary skill in the art. Such a person would understand the claims in light of the required capabilities and architectural features above.

4. The specification set forth in the '193 patent has numerous inconsistencies in its terminology. Some inconsistencies concern the data hierarchy (e.g., methods, control information, component assemblies). Other examples include the description of a non-secure host event processing environment and the concept of containment.

The following further summarizes Professor Maier's opinions.

I. EXPLANATION OF U.S. PATENT NO. 6,253,193

A. Asserted Capabilities of the Virtual Distribution Environment

The '193 Patent describes a system that is asserted to be the first universal, distributed processing system for persistently controlling digital information. This system was given the name "Virtual Distribution Environment" or "VDE". As described in the Patent, VDE promised at least the following mandatory features:

1. the creation of a comprehensive data security and commerce world;

2. the ability to handle all types of digital works independent of computing platform, making it a single, general purpose solution in contrast to multiple, limited solutions;

3. flexible control mechanisms that can be applied to any granularity of content;

4. control mechanisms that are configurable by any user, not just the system designers or content providers; and

5. isolation of the system programs and protected works from the non-VDE world, preventing observation, interference, or removal from the VDE, except as permitted by the VDE control mechanisms.

Although these features are promised by the '193 Patent, this does not mean that they are necessarily achievable.

1. Comprehensive Data Security and Commerce World

According to the '193 Patent, VDE is described as being the only comprehensive solution in a world of limited solutions. VDE is described as an end-to-end solution for digital works that guarantees the authenticity, confidentiality and integrity of the works and the VDE mechanisms. These protections are promised to be effective against any unauthorized activity by a third party (i.e. a user other than the creator of the work) that has physical possession of the computing hardware and wishes to circumvent the protections.

VDE must provide the ability to control the distribution and usage of digital works as well as tracking, reporting, auditing and handling payment for the distribution and usage. Additionally, VDE must support multiple business models simultaneously, for example, time-based and volume-based charging for the same digital work or licensing digital works with or without added sub-licensing rights.

Only those systems that are members of the electronic commerce world can participate in VDE commerce transactions. Consequently, all transactions must occur between

member systems since there is no way to control digital works that are outside the boundaries of the VDE world.

2. General Purpose

According to the '193 Patent, the VDE system is the only rights management solution needed by its users because it is capable of handling and protecting all types of digital works, such as digital audio, digital video, software, digital cash, digital documents, electronic publications, etc. within a single rights management framework, whereas previous systems handled only limited subsets of information types. It further states that VDE can function within all types of electronic devices, from smart cards, pagers and telephones to supercomputers.

3. Flexible

According to the '193 Patent, the VDE system can manage protected works in arbitrarily sized data chunks, down to the smallest atomic element. The Patent distinguished prior art systems that used access controls that were limited to the file level or resource level. The VDE system is described as being able to meter, track, bill and audit the usage of these arbitrary data chunks in addition to controlling the access to those data chunks. For example, a consumer can be charged by the number of bytes downloaded or by the number of paragraphs printed. Additionally, each of these actions can be specified independently, such that two objects can be metered differently, but billed identically.

This flexibility allows two different users to be charged at different rates, for different granularities, and in different currencies for using the same digital work. The '193 Patent distinguished prior art systems that lacked this flexibility.

4. Controls Configurable by All Users

According to the '193 Patent, the VDE system protects a digital work from the instant it is placed under VDE control subject to the permissions provided by the object creator (or rights holder) at the same or at another VDE "secure node." (The nature of the "secure node" is discussed later.) From that moment, the digital work becomes encapsulated within a VDE container. Then, the creator must grant permissions for accessing and distributing the digital work within the VDE object as well as identify how the object can be handled by other users of the VDE world.

These other users can create additional VDE-based controls for this protected work. In general, these controls only impose additional restrictions on the VDE object because they cannot conflict with the creator's VDE controls (except in the limited case in which the creator allows his controls to be modified by other users.) Even the end user is permitted to add VDE controls to VDE objects that he has received.

VDE controls are said to be persistent in that become permanently associated with the protected work once they are received, and they cannot be removed or deleted except as permitted by so-called "senior" VDE controls.

5. System Isolation

According to the '193 Patent, VDE protected works can only be accessed using VDE-certified foundation hardware and software. As a fundamental requirement, the VDE

foundation must isolate the internal workings of the system from the user because the user is not trusted.

Each computing device in the VDE world constitutes a "secure node" that must provide a "protected processing environment" (PPE) composed of VDE-certified foundation hardware and software. Sensitive materials such as protected works, administrative information, control information, and VDE software components, are passed between the protected processing environments of secure nodes inside "secure containers" that shield the materials from outside observation and alteration while in transit or in storage. The PPE must also shield all processing of the materials inside the PPE and also prevent the materials or process state information from "leaving" the VDE except as authorized by VDE control information. If the system fails to keep a protected work secret, then it can be distributed freely from that point onward. If the system fails to prevent alteration, then the consumer may receive invalid information (e.g., a bad stock quote), the consumer may receive less value than that for which he bargained (e.g., digital cash token that has been devalued), or the consumer's computer may be damaged by malicious code (e.g., virus-infected software), just to name a few examples. If the system fails to prevent the materials or process state information from leaving, then it can be moved to a system outside the VDE control regime for examination, manipulation, replication, or analysis.

Electronic devices outside the VDE world do not incorporate the VDE foundation, and hence are not constrained by VDE protocols. Thus, protected works are not permitted to be in clear text form outside of the isolated and rigidly controlled protected processing environment.

To guarantee the isolation and integrity of the PPE, the VDE foundation software itself must be protected by storing it in a location that is inaccessible to the user or by encrypting it when it is stored at a location that can be observed by the user.

B. VDE Core Architecture

According to the '193 Patent, three constituent building blocks are necessary to implement the VDE world:

1. VDE Foundation Hardware and Software – installed throughout an infrastructure of interlinked computing devices, each of which is called a "secure node";
2. The VDE "Secure Container" – a mechanism for packaging protected works, control information, and administrative information; and
3. The VDE "Control" – a mechanism for defining the regimen for using protected information that is inside a secure container.

Both controls and protected works are transferred between secure nodes by means of the secure container mechanism. Secure containers can be opened (and the protected works used) only within the protected processing environment of a secure node by executing VDE controls that regulate and track such activity.

The proper combination of these three building blocks isolates internal processing from the untrusted user (by creating an unbypassable foundation of hardware and software); isolates protected works from the untrusted user (by placing them in a shielded data structure); and provides a control mechanism that will allow the untrusted user to make use of the protected works only under controlled conditions.

1. VDE Foundation Hardware/Software

The VDE foundation hardware and software must ensure that the competing interests of both the owner and user of protected works are respected. The owner has an interest in controlling the distribution of his digital works and in compelling the reporting and payment for such use. The user has an interest in the control of his computing device, his privacy, and the availability of digital works for which he has paid.

The VDE foundation hardware and software must provide a sequestered venue in which external authority dominates the user's local authority in the control of information and processing. This VDE foundation hardware and software is the basis for any VDE installation on a device

A VDE secure node is a device that provides a VDE installation incorporating VDE foundation hardware and software as the base stratum on which all VDE functions are executed. In any secure node where protected works are used or where VDE control information is created or modified, a VDE secure subsystem core must be present. This core is enclosed by a "tamper resistant security barrier" that prevents observation of, interference with, and leaving of information and processes except as authorized by VDE control information.

This VDE secure subsystem core handles encrypting and decrypting data and code, storing control and metering information, managing secure communication with other VDE secure subsystem cores at other secure nodes, dynamically assembling and executing VDE control procedures, and updating control information for protected works.

Control procedures for the promised permission checking, metering, billing, and budget management features all execute within the VDE secure subsystem core.

The VDE foundation hardware and software must guarantee that control procedures triggered by user or system events are executed correctly and completely in the VDE secure subsystem core. Both correctness and completeness are necessary to preserve the integrity of VDE control regime. Failure can compromise the rights, privacy, or financial interests of the owner or user of the protected works.

According to the '193 Patent, these functions are provided and enforced by a secure processing unit (SPU) that is protected by a special purpose physical enclosure (the tamper resistant security barrier) that conceals the underlying VDE processing from observation or interference by external persons or processes, and that destroys information rather than allow the information to leave the VDE subsystem core via unauthorized means.

The '193 Patent suggests that a tamper resistant security barrier might be simulated solely in software by using several known software techniques, but it gives no specific direction as to how these techniques can be applied to achieve the guarantees required by the VDE secure subsystem core in an environment that is under the control of an untrusted user.

2. VDE Secure Containers

An invariant requirement of the VDE container concept is that no access or use can be made of the protected works within a VDE container except as regulated by associated VDE control information. This associated control information can be provided in the

same secure container that holds the protected works or it can be provided independently in a separate secure container.

In addition to the protected works included within a secure container, there can be references to other digital works stored external to the container. However, the container cannot regulate other access or usage to these externally stored digital works. ("Containment" is discussed further is Section IV. D.)

VDE secure containers can contain administrative information, such as auditing, tracking, and billing requests and reports.

The internal structure of a VDE secure container must be able to store independently manageable digital works. Subsections of a VDE secure container can be encrypted by different keys, including subdivisions of a single digital work.

The internal structure of a VDE secure container must be able to store other VDE secure containers nested inside it. Each nested container is subject to its own independent control information. Control information corresponding to the outer container may not override more restrictive control information that corresponds to a secure container nested within it.

The VDE secure container supports modification of its contents and its control information subject to the current corresponding control information.

Because of this capability, a VDE secure container may be empty in the sense that it does not contain a digital work while it does contain control information that identifies the digital work that can be added to the secure container. Thus, a VDE secure container can be used as a mobile agent to retrieve digital works from remote locations.

3. VDE Controls

According to the '193 Patent, the configurability and flexibility of the VDE system arises jointly from the modular and independently selectable nature of control information and the dynamic construction and execution of control procedures within the VDE secure subsystem of a computing device. As used herein, the VDE secure subsystem refers to the VDE foundation hardware and software residing within the tamper resistant security barrier.

VDE controls are executable procedures constructed by the VDE foundation as a response to a request to access or use a specific protected work. The control is constructed inside the VDE secure subsystem using VDE control information. VDE control information is composed of executable code, rule information that is enforced by the executable code, and blueprint instructions for constructing the executable control. The VDE secure subsystem guarantees that the control procedure is constructed according to the blueprint instructions and that the components used in the construction are authentic as to source, identity, and data integrity.

All use of protected works is regulated by corresponding control information that is used to construct each executable control procedure. Different control procedures can regulate auditing, billing, metering, tracking and usage events (such as printing, rendering, copying, etc.) with respect to individual users for a single instance of a protected work. These events cannot occur except as regulated by the execution of the individual control procedures. Additionally, these control procedures can be applied at arbitrarily fine levels of granularity, such as charging for the number of bytes read.

Any VDE user can define control procedures to the extent permitted by senior VDE control information.

Control information is deliverable independent of the protected work. Individual portions of control information are deliverable independent of each other. Control information made by added, modified, or replaced over time to the extent permitted by earlier control information. Because independent control information for any given instance of a protected work can be created by different sources at different locations and different times, the control information from these sources can be in conflict. VDE must supply a means for resolving these conflicts. According to the '193 Patent, the executable controls negotiate to determine the conditions under which a protected work may be used. Thus, controls are said to "evolve" over time.

Once delivered to a VDE node with the corresponding protected work, control information persists throughout the life of the protected work.

The VDE controls must support a broad range of control regimes, all of which can co-exist on a single VDE secure node.

Dynamic assembly and execution of a VDE control must occur within the VDE secure subsystem. Construction of a VDE control from its component parts in a non-VDE system allows unconstrained access to digital works. Thus, VDE control information is transmitted between secure nodes using VDE secure containers and stored at VDE nodes in encrypted form whenever outside the VDE secure subsystem.

Executable control procedures are constructed from load modules, data, and VDE methods. These control procedures are assembled and executed in response to user and

system events. According to some statements in the '193 Patent, a "component assembly" is a VDE control procedure.

C. Claim Interpretation

A person of ordinary skill in the art would understand the claims of the '193 Patent in light of the mandatory capabilities and architectural components described above.

D. Summary of Internal Inconsistencies.

The '193 Patent contains numerous internal inconsistencies. Examples of these inconsistencies are given below.

1. Use of Quotations

Hundreds of terms are set off in quotations throughout the specification. These terms include: detail description, virtual distribution environment, electronic highway, VDE aware, content, virtual, things, chain of handling and control, rules and controls, CD ROM, information utility, switch, transaction processor, usage analyst, operating system, method, budget, atomic, firmware, hash bucket, peripheral device, event-based, multi-threaded, locking, Remote Procedure Call, two-phase commit, and read only. Some of these terms are coined (such as VDE aware; rules and controls; and usage analyst) while many are well known computer concepts (such as operating system and Remote Procedure Call.).

In many cases, it is unclear whether any particular use of quotation marks was intended to introduce a coined term, to indicate figurative or metaphorical usage of a term, to indicate non-standard or a weakened usage of a term, or something else

2. System Availability

In the Absract, the '193 Patent asserts that "the invention . . . maintain[s] the integrity, availability, and/or confidentiality" of protected works. However, the system described does not appear to be designed to guarantee the availability of protected works. Rather, any deviation from the expected processing sequence is considered to be evidence of an attempt to crack the system or steal the protected works. In response, the system is likely to halt all processing until a trusted VDE administrator intervenes and resets the system. Additionally, the '193 Patent uses denial of service to enforce reporting obligations imposed by a rights holder. This practice is not consistent with preserving availability of digital works.

3. "Container" vs. "Object"

There is no consistent delineation in the '193 Patent between the terms "container" and "object." Initially, there appears to be a distinction in that the container is a shell data structure that is encapsulating data and the object is the combination of the container data structure and the encapsulated data. See Fig. 5A. Elsewhere, this distinction is blurred by the use of such phrases as:

"secure object (content container)";

"VDE content container is an object"; and

"VDE container (object)",

which appear to make container and object synonymous.

4. The Property of Being "Contained"

In the '193 Patent, there is no clear definition for the term "contain." The '193 patent states at one point that a container such as "container 302 may 'contain' items without those items actually being stored in the container." This definition of "contain" to include "referencing" is not customary in information storage terminology.

Subsequent examples in the '193 indicate that "contain" and "reference" are distinct relationships. For example, "may contain or reference" is used numerous times such as in "Load modules 1100 may contain or reference other load modules." and as in "Container 300y may contain and/or reference. . . ."

5. Inconsistent Data Structure Hierarchy

The hierarchy and relationships amongst rules, controls, methods, load modules, control information, and other data structures is inconsistent.

a) "Rules and Controls" vs. "Control Information"

The term "control information" is defined in the "Background and Summary of the Invention" of the '193 Patent as: ". . . load modules, associated data and methods . . ." Later, the specification uses the phrase " 'rules and controls' (control information)" as if the phrases "control information" and "rules and controls" are synonymous. Further, it states that "rules and controls" can be in the form of: "a 'permissions record' 808; 'budgets' 308 and 'other methods' 1000", but makes no mention of load modules.

Subsequent uses of "control information" such as: ". . . other aspects of the information to be contained within the object (e.g., rules and control information, identifying

information, etc.)"; and "the user may specify permissions, rules and/or control information." indicate that rules are different and distinct from control information.

b) "Component Assembly" vs. "Control Information"

In the '193 Patent, the relationship between component assembly and control information in the data hierarchy is defined inconsistently. Contrast the statement:

> "In this example control information may include one or more component assemblies that describe the articles within such a container (e.g. one or more event methods referencing map tables and/or algorithms that describe the extent of each article)."

with:

> ". . . control information (*typically a collection of methods related to one another by one or more permissions records, including any method defining variables*) . . ." [italics in original]
>
> "This "channel 0" "open channel" task may then issue a series of requests to secure database manager 566 to obtain the "blueprint" for constructing one or more component assemblies 690 to be associated with channel 594 (block 1127). In the preferred embodiment, this "blueprint" may comprise a PERC 808 and/or URT 464.'

In one case, the component assembly is a part of control information, but in the other case, control information is separable from and describes how to build a component assemblies.

c) "Budgets"

According to the '193 Patent, "budgets" are a special type of "method". Methods are defined as containing, among other things, "User Data Elements". Elsewhere, budgets are cited as a common type of User Data Element. This inconsistency creates confusion as to whether any given use of the term "budget" refers to an executable method or a non-executable data structure.

6. "Load Module"

According to the '193 Patent, executable code is provided in the form of "'atomic' load modules", presumably meaning that they are the smallest unit of executable code. Later, however, load modules are sub-dividable into smaller load modules, which is inconsistent with atomicity.

7. The "Non-Secure" "Protected Processing Environment"

According to the '193 Patent, a necessary feature of a VDE computer is the "protected processing environment" or "PPE". Secure Event Processing Environments ("SPE"), in which all sensitive processing is handled inside a hardware device called a Secure Processing Unit ("SPU") are stated as being one type of PPE. Host Event Processing Environments ("HPE") are also said to be a type of PPE. The HPE classification is further described as having two sub-types: secure and non-secure. Additionally, the specification defines the three abbreviations as synonymous and interchangeable starting at column 103 of the specification, unless the context of any given passage indicates otherwise.

Further, no criteria are provided for distinguishing between a "secure HPE" and a "non-secure HPE". Thus, it is not possible to reconcile the "non-secure HPE" as a secure operating environment or protected processing environment.